REMARKS

Reconsideration of this application as amended is respectfully requested.

In the Office Action, claims 1-7, 9-11, 13, 15-22 and 24-35 are pending and rejected. Claims 1, 11, 20, and 29 have been amended. Claim 36 has been added. No new matter has been added.

Claim Objections

Claims 20 is objected to because of informalities in the Office Action and has been amended as suggested. The Applicants respectfully request the objections towards the claim to be withdrawn.

Claim Rejection – 35 U.S.C. §101

Claims 20-22 and 24-28 are rejected under 35 U.S.C. §101. Paragraph [0037] in the Specification has been amended to present the scope of a machine-readable storage medium. The Applicants submit that the claim 20 and its dependent claims 21-22, 24-28 are directed to the statutory matters as required by 35 U.S.C. §101. The Applicants respectfully request the rejections towards the claims to be withdrawn.

Claim Rejection – 35 U.S.C. §103

The following discussion sets forth in detail Applicants' analysis with respect to the patentability of claims 1-7, 9-10 and 29-35.

A. It is asserted in the Office Action that claims 1-7, 9-10 and 29-35 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy et al. ("Slipstream Processors: Improving both Performance and Fault Tolerance", ASPLOS, pp. 257-268, Nov. 2000) ("Sundaramoorthy") in view of U. S. Patent No. 6,757,811 issued to Mukherjee

("Mukherjee") in view of Hennessy and Patterson ("Computer Architecture A Quantitative Approach", Morgan Kaufmann, 1996) ("Hennessy"). Applicants respectfully traverse the aforementioned rejections for the following reasons.

According to MPEP §2143.03, "*All words in a claim must be considered* in judging the patentability of that claim against the prior art." (In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970), emphasis added.)

Claim 1, as amended, contains the limitations of

> 1. An apparatus comprising:
> a first processor and a second processor each having a scoreboard and a decoder;
> a plurality of memory devices coupled to the first processor and the second processor;
> **a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer and is operable to transfer register values from the second processor to the first processor**;
> a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and
> a plurality of memory instruction buffers coupled to the first processor and the second processor;
> **wherein the first processor and the second processor perform single threaded applications using multithreading resources, and the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and the single threaded application executed on the second processor avoids branch mispredictions from information received from said first processor**. (emphasis added)

Applicants submit that Sundaramoorthy, Hennessy, and Mukherjee fail to disclose at least the two limitations of claim 1.

Sundaramoorthy discloses a multiprocessor system that executes two (i.e., multiple streams/threads) pseudo-redundant programs on separate processors on the same chip. The two programs have a differing amount of instructions. (Sundaramoorthy, column 2, lines 34-56). That is, one of the programs has more instructions than the other. (Sundaramoorthy, page 258, first column, lines 40-55). Sundaramoorthy further discloses that the A-stream has fewer instructions than the R-stream, which receives information from the A-stream. Both programs run in parallel on two processors. Sundaramoorthy fails to disclose "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

In other words, Sundaramoorthy executes multiple-streams on two separate processors where each stream has a different number of instructions, as opposed to Applicants' claimed invention, which executes a single thread application without converting or duplicating the single thread application to a multiple-thread application when using multiple-thread processing resources, where the single thread application is executed on two processors without changing the number of instructions (i.e., an exact duplicate).

Hennessy discloses using score boarding to aid in allowing instructions to execute out of order. Sundaramoorthy, however, is directed to finding instructions that do not effect final program output and removes these instructions from a second stream, such as, for example redundant instructions. Therefore, the combination of the two prior art documents would not result in Applicants' claimed invention. Further, Hennessy fails to disclose "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

Mukherjee is relied upon for disclosing a multi-threaded processor executes two single threads, where one thread executes slightly ahead of the other for fault tolerance. The two threads are duplicates of one another. Mukherjee deals with multiple threads in a multi-threading processor, not single threaded processes. Mukherjee further asserts that the leading and trailing thread are executed on the same processor. Mukherjee fails to disclose

> the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Additionally, claim 1 requires "a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer and is operable to transfer register values from the second processor to the first processor", where the first processor executes a single threaded application ahead of the second processor as recited in claim 1. The Office Action alleges that "a first buffer" is disclosed by the delay buffer in Sundaramoorthyin (Fig. 1 and col. 10, lines 17-21). Applicants respectfully disagree. Sundaramoorthy states that "the delay buffer is a simple FIFO queue that allows the A-stream to communicate control flow and data flow outcomes to the R-stream." Sundaramoorthy defines that "the leading program is called the advanced stream, or A-stream, and the trailing program is called the redundant stream, or R-stream." (col. 2, lines 4-6). Sundaramoorthy discloses a delay buffer that allows the advance stream to communicate control flow and data flow from A-stream to R-stream (not the other way around). In short, a delay buffer is not the first buffer as claimed in claim 1. Sundaramoorthy fails to disclose the limitation as required. Mukherjee and Hennessy also fail to cure this deficiency of Sundaramoorthy.

Moreover, it is asserted in the Office Action that it would benefit Sundaramoorthy to run two streams having the same amount of instructions with one running ahead of the other. This teaches away the disclosure of Sundaramoorthy because one of ordinary skill in the art would know that streams of the same amount of instructions only adds latency. It is noted that, Sundaramoorthy speculatively creates a shorter version of the program to exploit computation that influences highly-predictable branches. Sundaramoorthy also states that A-stream is sped up because it is shorter. As stated in MPEP 2143.01 (VI), the proposed modification can not change the principle of operation of a reference. "*If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious.*" Applicants submit that the combination proposed in the Office Action would change the principle of Sundaramoorthy. Therefore, the combination fails to establish a prima facie case with respect to claim 1.

Furthermore, by viewing the disclosures of Sundaramoorthy, Mukherjee and Hennessy, one cannot jump to the conclusion of obviousness without impermissible hindsight. Applicants submit that without first reviewing Applicants' disclosure, no thought would have been made to the limitations of "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

Even if Sundaramoorthy were combined with Mukherjee and Hennessy, the resulting invention would still not include all of Applicants' claimed limitations. Therefore, claim 1 is not obvious over Sundaramoorthy in view of Mukherjee and Hennessy.

Claim 29 recites at least the same limitations as discussed above and is therefore not obvious over Sundaramoorthy in view of Mukherjee and Hennessy. The claims that directly or indirectly depend from amended claims 1 and 29, namely claims 2-7 and 9-10, and 30-35, respectively, would also not be obvious over Sundaramoorthy in view of Mukherjee and Hennessy for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 1-7, 9-10 and 29-35 are respectfully requested.

The following discussion sets forth in detail Applicants' analysis with respect to the patentability of claims 11, 13, and 15-19.

B. It is asserted in the Office Action that claims 11, 13, and 15-19 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy and in further view of Akkary (WO 99/31594). Applicants respectfully traverse the aforementioned rejection for the following reasons.

Claim 11, as amended, recites:

> 11. A method comprising:
> executing a plurality of instructions in a single thread by a first processor;
> executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction;
> tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor,
> transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information;
> transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and
> clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer,
> wherein the first processor and the second processor execute single threaded applications using multithreading resources, said single thread is not converted to an explicit multiple-thread application, said single thread contains the same number of instructions when executed on said first processor and said second processor, and the single thread executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicants' claims 13 and 15-19 directly depend on amended claim 11. Applicants' claim 11 contains similar limitations as claim 1 above. Namely,

> executing a plurality of instructions in a single thread by a first processor; executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor, transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information; transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, said single thread is not converted to an explicit multiple-thread application, said single thread contains the same number of instructions when executed on said first processor and said second processor, and the single thread executed on the second processor avoids branch mispredictions using information received from said first processor. (emphasis added)

Based at least from the foregoing reasons with respect to claim 1, neither Sundaramoorthy, Mukherjee, Hennessy, and therefore, nor the combination of the three disclose the limitations contained in claim 11.

Akkary discloses a system for ordering loads and stores in a multi-threaded processor using load and store buffers. Applicants are well aware of Akkary as Akkary is owned by Applicants' Assignee. Akkary does not disclose

> the first processor and the second processor execute single threaded applications using multithreading resources, said single thread is not converted to an explicit multiple-thread application, said single thread contains the same number of instructions when executed on said first processor and said second processor, and said single thread executed on the second

> processor avoids branch mispredictions using information received from said first processor.

Therefore, combining Akkary with Sundaramoorthy, Mukherjee and Hennessy would still not result in all limitations set forth above.

Additionally, claim 11 requires "executing said plurality of instructions in the single thread by a second processor as directed by the first processor". The Office Action alleges that IR-predictor and IR-detector in figure 1 (Sundaramoorthy) is part of the first processor i.e. R-stream processor. Applicants respectfully disagree. Sundaramoorthy shows that IR-predictor and IR-detector are part of slipstream components but not in a single processor (Table 2, page 264). Other cited references also fail to cure this deficiency.

Neither Sundaramoorthy, Hennessy, Mukherjee, Akkary, and therefore, nor the combination of the four, disclose the limitations contained in claim 11, as listed above. Since neither Sundaramoorthy, Hennessy, Mukherjee, Akkary, nor the combination of the four, disclose all the limitations of Applicants' claim 11, Applicants' claim 11 is not obvious over Sundaramoorthy in view of Mukherjee, Hennessy, and further in view of Akkary since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly depend from claim 11, namely claims 13 and 15-19, would also not be obvious over Sundaramoorthy in view of Mukherjee, Hennessy and further in view of Akkary for at least the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 11, 13, and 15-19 are respectfully requested.

The following discussion sets forth in detail Applicants' analysis with respect to the patentability of claims 20-22, and 24-28.

C. It is asserted in the Office Action that claims 20-22, and 24-28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy in view of Akkary, as applied above, and in further view of Tanenbaum "Structured Computer Organization," Prentice-Hall, 1984, pp. 10-12 ("Tanenbaum"). Applicants respectfully traverse the aforementioned rejections for the following reasons.

Applicants' claim 20 contains the limitations of

> 20. An apparatus comprising a machine-readable storage medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:
>
> executing a single thread by a first processor;
>
> executing said single thread from a second processor as directed by the first processor, the second processor executing instructions ahead of the first processor to avoid misprediction;
>
> tracking at least one register that is one of loaded from a first buffer, and written by said second processor, said tracking executed by said second processor, the first buffer being a register file buffer, and
>
> clearing a store validity bit and setting a mispredicted bit in a load entry in a second buffer if a replayed store instruction has a matching store identification (ID) portion, the second buffer being a trace buffer,
>
> wherein the first processor and the second processor execute single threaded applications using multithreading resources, said single thread is not converted to an explicit multiple-thread application, said single thread contains the same number of instructions when executed on said first processor and said second processor, and said single thread executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant has discussed similar limitations above in section (B) regarding Sundaramoorthy in view of Mukherjee, Hennessy, and Akkary. Tanenbaum is relied on for asserting that "any instruction executed by hardware can also be executed in software." Tanenbaum fails to cure the deficiency of Sundaramoorthy, Mukherjee, Hennessy, and Akkary with respect to claim 11. Therefore, combining Tanenbaum with Akkary, Sundaramoorthy, Mukherjee and Hennessy would still not result in Applicants' claim 20.

Neither Sundaramoorthy, Mukherjee, Hennessy, Akkary, Tanenbaum, nor the combination of the five, discloses the limitations contained in Applicants' claim 20, as listed above. Since neither Sundaramoorthy, Mukherjee , Hennessy, Akkary, Tanenbaum, nor the combination of the five disclose all the limitations of Applicants' claim 20, Applicants' claim 20 is not obvious over Sundaramoorthy in view of Mukherjee, Hennessy and Akkary and further in view of Tanenbaum since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly or indirectly depend from claim 20, namely claims 21-22 and 24-28, would also not be obvious over Sundaramoorthy in view of Mukherjee, Hennessy and Akkary, and further in view of Tanenbaum for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 20-22, and 24-28 is respectfully requested.

D. Improper Official Notice

In the Office Action, many claim limitations are deemed as inherently disclosed where Official Notice is taken in the absence of support from publications. The Office Action fails to show a reference that discloses "the register buffer comprising an integer register buffer and a predicate register buffer" as required in claim 10. The Office Action also fails to show a reference that discloses "clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer" as required in claim 11. Moreover, all limitations are required in claim 17 and claim 18 are deemed as inherent without support from publications. Applicants further submit the limitations are not inherently or explicitly disclosed by references cited in the Office Action especially when the claimed invention is considered as a whole. Applicants respectfully request references be provided to support the rejections or rejections of the claims be withdrawn.

New Claim

Applicants present a new dependent claim (claims 36) depending from claim 29. In view of the foregoing reasons, claim 29 is allowable and therefore the dependent claim 36 is also allowable over the cited references. Additionally, claim 36 requires "wherein the second processor is operable to commit results in one commit cycle based at least on the information received from the first processor". The cited references fail to teach at least the limitation as set forth in the claim above. Therefore, at least for the foregoing reason, Applicants submit that claim 36 is allowable. Applicants respectfully request the allowance for claim 36.

CONCLUSION

Applicants respectfully request the amendments of claims to be admitted. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call the undersigned attorney at (408) 720-8300.

Please charge Deposit Account No. 02-2666 for any shortage of fees in connection with this response.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 1/7/09

Michael J. Mallie
Attorney for Applicant
Reg. No. 36,591

1279 Oakmead Parkway
Sunnyvale, California 94085-4040
(408) 720-8300